Feature-driven Video Segmentation and Advanced Querying with vitrivr-engine

Luca Rossetto[1][0000−0002−5389−9465] and Ralph Gasser[2][0000−0002−3016−1396]

[1] Dublin City University, Dublin, Ireland
luca.rossetto@dcu.ie
[2] University of Basel, Basel, Switzerland
ralph.gasser@unibas.ch

Abstract. The vitrivr system has been a long-term contributor to the annual Video Browser Showdown. In our 2024 participation, we introduced a new core retrieval engine to the vitrivr stack, dubbed the *vitrivr-engine*. For the 2025 edition, we extend the engine with mechanisms for feature-driven video segmentation and additional querying means. We also extend the compatibility to additional storage backends, which enables *vitrivr-engine* to leverage advances made in the area of vector database systems.

Keywords: Interactive Video Retrieval · Feature-based Video Segmentation · Advanced Retrieval Query Processing

1 Introduction

The *vitrivr* system is a general-purpose multimedia retrieval stack and a long-time contributor to the Video Browser Showdown. For most of these participations to VBS, it used the open-source query processing and feature extraction engine *Cineast* [9] as a core component. In 2024, we introduced Cineast's successor, a system we call the *vitrivr-engine* [1]. While this deployment of vitrivr used a comparatively minimalist user interface, based on [12], we adapted our customary user interface *vitrivr-ng* [3] to be compatible with *vitrivr-engine*'s new API in our contribution [11] to the 2024 Lifelog Search Challenge [4].

In this year's participation, we build upon *vitrivr-engine*'s modularity and flexibility to introduce new functionality. Specifically, we add a feature-specific video segmentation mechanism and extend the querying workflow to support more complex query types. In addition, we extend the mechanisms for communicating with storage backends to make *vitrivr-engine* more backend agnostic.

2 Contributions

The contributions made to *virivr-engine* in the context of this participation to VBS 2025 are threefold: First, we introduce mechanisms for segmenting video based on properties of extracted features – rather than the other way around –

and add the necessary facilities to deal with the resulting consequences. Second, we extend the vector-space querying workflow beyond k-nearest-neighbor operations. Third, we extend the mechanisms to interact with the storage backend, making *vitrivr-engine* more backend agnostic, which also enables us to leverage functionality offered by different vector databases. These contributions are explained in more detail below.

2.1 Feature-driven Video Segmentation

In general-purpose video retrieval, the videos are commonly too long and semantically diverse to serve as an atomic unit of retrieval. They are instead segmented into smaller units, commonly referred to as *shots*, that are more uniform in terms of content. Such segmentation can be achieved in various ways. The simplest approach, in the absence of any further identifying information, is just to segment the videos equifrequently into segments with a fixed length. More sophisticated approaches are based on the actual content of the frames and use anything from low-level color properties to higher-level semantic content, e.g., [13], to detect suitable segment boundaries.

Two out of the three datasets used for VBS 2025 – V3C [10] and MVK [15] – already come with automatically generated segmentation information. The former uses boundaries generated using localized color histograms, while the latter uses a fixed-length segmentation scheme using a one-second interval.

Since, for an appropriately segmented shot, all frames within a shot are very similar to each other, it is often considered to be sufficient to use a single frame as a representative for the entire shot. This effectively reduces the video retrieval problem to an image retrieval problem and opens up access to a larger range of retrieval features. In recent years, CLIP [7] and its variants have emerged as the most popular choice in this regard. This works well if the notion of frame similarity of the feature and the one of the video segmented are well-aligned. If there is a misalignment, however, it can lead to a selection of representative frames that are not actually representative of the entire shot, thereby causing certain parts of a video to not be retrievable. This issue will be exacerbated when multiple features with different similarity notions are used on the same segments, as they can't possibly be well-aligned with all features.

To overcome this possible mismatch, we introduce a dynamic, per-feature segmentation scheme based on the feature vectors themselves. We sample the video at a fixed frame rate, usually below the effective video playback frame rate to reduce computational load, and compute the feature representation for each frame. The first feature vector serves as an anchor, and all subsequent vectors are compared to it. If the distance between these two vectors exceeds a pre-determined, feature-specific threshold, the vector is considered to no longer adequately represent the previous ones and a new segment is started. All previous frames will be assigned to a segment, and the latest vector will again serve as an anchor for the next segment.

Figure 1 illustrates the effect of such an approach by showing the pairwise Cosine distances between CLIP features extracted from the first five V3C videos

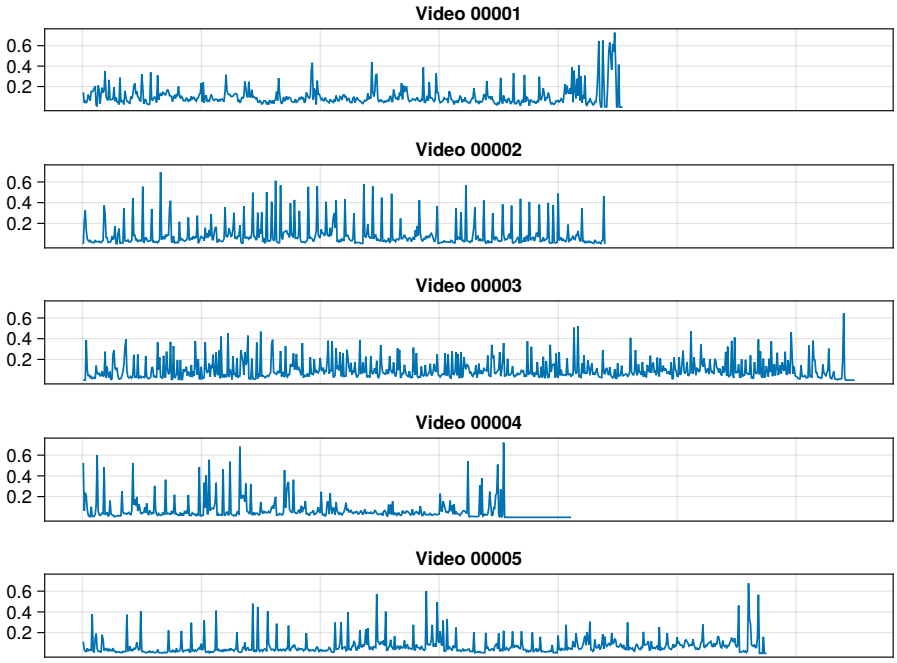

Fig. 1. Cosine distances between CLIP features extracted from consecutively sampled frames with a framerate of two frames per second for the first five videos of the V3C dataset. The horizontal axis shows the video time, and the vertical axes show the distance values.

sampled at a constant rate of two frames per second. The peaks in the plotted lines correspond to transitions where the difference between two frames is very large, i.e., CLIP considers them to be quite dissimilar. This indicates suitable segment boundaries for this feature Other feature representations might show peaks in different positions, benefiting from different segment boundaries.

While this segmentation scheme reduces the segmentation error per feature, it complicates result aggregation across multiple features. Being able to use multiple features with different segment boundaries requires the intersection of segments, possibly leading to an increase in segments being returned to the user. It is, therefore, important to carefully consider how results are to be combined when issuing a query.

2.2 Extended Querying Workflow

One of the novelties in *vitrivr-engine* is its flexible data flow architecture used for query specification and execution. While this data flow was already capable of performing arbitrary operations on retrieved results, in this iteration, we extend it to also be capable of performing operations on the input data. Specifically,

operations can be performed on user-defined inputs, including document IDs, feature vectors, text queries, reference images, etc. These operations can be used to transform or aggregate such user-specified input data. A simple example would be the lookup of a known vector based on an ID, which can be used for a subsequent query.

A more elaborate effect can be achieved by aggregating such inputs in various ways. Inspired by approaches such as the one used by Exquisitor [8], we add an operator to compute a linear SVM based on known vector representations, together with a user-provided positive and negative label for each. The result can be used in a *hyperplane* query, which retrieves the vectors *furthest* from a given hyperplane, which is equivalent to a k furthest neighbor query using the inner product distance. Such queries allow for expressive relevance feedback.

2.3 Storage Backend Agnosticism

In the context of the vitrivr project, we developed several database management systems capable of handling vector data in addition to more traditional data types supported by relational database management systems. The latest of these systems is Cottontail DB [2], which has been used for all participations to VBS since 2020. Since then, the area of vector databases has enjoyed increased interest, and many existing DBMSs have been extended by vector functionality. To make use of such efforts, we extend the storage interface mechanism used by *vitrivr-engine* to support existing vector databases outside the vitrivr ecosystem. Specifically, we add support for PostgreSQL, which can be used as a vector database via its extensions pgvector[3] and pgvectorscale.[4] The former comes with support for vector index structures such as HNSW [5] and IVFFlat [6], while the latter brings support for an optimized version of DiskANN [14].

While PostgreSQL does not support all the vector operations available in Cottontail DB, it is powerful enough to handle the most commonly used types of vector comparison operations required by *vitrivr-engine*. Since PostgreSQL is, however, used in a wide range of productive applications, it is much more performance-optimized when compared to our research system, Cottontail DB. Being able to choose a suitable storage backend, depending on a specific application, therefore, greatly increases the flexibility and applicability of *vitrivr-engine*. The efforts made to enable the use of PostgreSQL in *vitrivr-engine* also laid the foundations to support additional storage backends in the future.

3 Conclusion

In this paper, we have presented an overview of the additions made to vitrivr for its participation in the 2025 Video Browser Showdown. We introduced a feature-driven video segmentation mechanism to eliminate possible mismatches

[3] https://github.com/pgvector/pgvector
[4] https://github.com/timescale/pgvectorscale

between segmentation measures and feature mechanisms and extended the already flexible query processing pipeline by means of performing operations on input data. We also extended the capabilities of *vitrivr-engine* with respect to supported storage backends, making it more backend agnostic.

Acknowledgments. This work was partly supported by the Swiss National Science Foundation through the project MediaGraph (contract no. 202125), and the Horizon Europe project XR mEdia eCOsystem (XReco), based on a grant from the Swiss State Secretariat for Education, Research and Innovation (contract no. 22.00268).

References

1. Gasser, R., Arnold, R., Faber, F., Schuldt, H., Waltenspül, R., Rossetto, L.: A new retrieval engine for vitrivr. In: Rudinac, S., Hanjalic, A., Liem, C.C.S., Worring, M., Jónsson, B.Þ., Liu, B., Yamakata, Y. (eds.) MultiMedia Modeling - 30th International Conference, MMM 2024, Amsterdam, The Netherlands, January 29 - February 2, 2024, Proceedings, Part IV. Lecture Notes in Computer Science, vol. 14557, pp. 324–331. Springer (2024). https://doi.org/10.1007/978-3-031-53302-0_28
2. Gasser, R., Rossetto, L., Heller, S., Schuldt, H.: Cottontail DB: an open source database system for multimedia retrieval and analysis. In: Chen, C.W., Cucchiara, R., Hua, X., Qi, G., Ricci, E., Zhang, Z., Zimmermann, R. (eds.) MM '20: The 28th ACM International Conference on Multimedia, Virtual Event / Seattle, WA, USA, October 12-16, 2020. pp. 4465–4468. ACM (2020). https://doi.org/10.1145/3394171.3414538
3. Gasser, R., Rossetto, L., Schuldt, H.: Towards an all-purpose content-based multimedia information retrieval system. CoRR **abs/1902.03878** (2019), http://arxiv.org/abs/1902.03878
4. Gurrin, C., Zhou, L., Healy, G., Bailer, W., Dang-Nguyen, D., Hodges, S., Jónsson, B.Þ., Lokoc, J., Rossetto, L., Tran, M., Schöffmann, K.: Introduction to the seventh annual lifelog search challenge, lsc'24. In: Gurrin, C., Kongkachandra, R., Schoeffmann, K., Dang-Nguyen, D., Rossetto, L., Satoh, S., Zhou, L. (eds.) Proceedings of the 2024 International Conference on Multimedia Retrieval, ICMR 2024, Phuket, Thailand, June 10-14, 2024. pp. 1334–1335. ACM (2024). https://doi.org/10.1145/3652583.3658891
5. Malkov, Y.A., Yashunin, D.A.: Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. IEEE Trans. Pattern Anal. Mach. Intell. **42**(4), 824–836 (2020). https://doi.org/10.1109/TPAMI.2018.2889473
6. Mallia, A., Khattab, O., Suel, T., Tonellotto, N.: Learning passage impacts for inverted indexes. In: Diaz, F., Shah, C., Suel, T., Castells, P., Jones, R., Sakai, T. (eds.) SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021. pp. 1723–1727. ACM (2021). https://doi.org/10.1145/3404835.3463030
7. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event. Proceedings of Machine Learning Research, vol. 139, pp. 8748–8763. PMLR (2021), http://proceedings.mlr.press/v139/radford21a.html

8. Ragnarsdóttir, H., Þorleiksdóttir, Þ., Khan, O.S., Jónsson, B.Þ., Guðmundsson, G.Þ., Zahálka, J., Rudinac, S., Amsaleg, L., Worring, M.: Exquisitor: Breaking the interaction barrier for exploration of 100 million images. In: Amsaleg, L., Huet, B., Larson, M.A., Gravier, G., Hung, H., Ngo, C., Ooi, W.T. (eds.) Proceedings of the 27th ACM International Conference on Multimedia, MM 2019, Nice, France, October 21-25, 2019. pp. 1029–1031. ACM (2019). https://doi.org/10.1145/3343031.3350580
9. Rossetto, L., Giangreco, I., Schuldt, H.: Cineast: A multi-feature sketch-based video retrieval engine. In: 2014 IEEE International Symposium on Multimedia, ISM 2014, Taichung, Taiwan, December 10-12, 2014. pp. 18–23. IEEE Computer Society (2014). https://doi.org/10.1109/ISM.2014.38
10. Rossetto, L., Schuldt, H., Awad, G., Butt, A.A.: V3C - A research video collection. In: Kompatsiaris, I., Huet, B., Mezaris, V., Gurrin, C., Cheng, W., Vrochidis, S. (eds.) MultiMedia Modeling - 25th International Conference, MMM 2019, Thessaloniki, Greece, January 8-11, 2019, Proceedings, Part I. Lecture Notes in Computer Science, vol. 11295, pp. 349–360. Springer (2019). https://doi.org/10.1007/978-3-030-05710-7_29
11. Sauter, L., Gasser, R., Rettig, L., Schuldt, H., Rossetto, L.: General purpose multimedia retrieval with vitrivr at lsc'24. In: Proceedings of the 7th Annual ACM Workshop on the Lifelog Search Challenge, LSC 2024, Phuket, Thailand, 10 June 2024. pp. 47–52. ACM (2024). https://doi.org/10.1145/3643489.3661120
12. Sauter, L., Schuldt, H., Waltenspül, R., Rossetto, L.: Novice-friendly text-based video search with vitrivr. In: Chetouani, A., Bailer, W., Gurrin, C., Benoît, A. (eds.) 20th International Conference on Content-based Multimedia Indexing, CBMI 2023, Orleans, France, September 20-22, 2023. pp. 163–167. ACM (2023). https://doi.org/10.1145/3617233.3617262
13. Soucek, T., Lokoc, J.: Transnet V2: an effective deep network architecture for fast shot transition detection. CoRR **abs/2008.04838** (2020), https://arxiv.org/abs/2008.04838
14. Subramanya, S.J., Devvrit, Simhadri, H.V., Krishnaswamy, R., Kadekodi, R.: Rand-nsg: Fast accurate billion-point nearest neighbor search on a single node. In: Wallach, H.M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E.B., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada. pp. 13748–13758 (2019), https://proceedings.neurips.cc/paper/2019/hash/09853c7fb1d3f8ee67a61b6bf4a7f8e6-Abstract.html
15. Truong, Q., Vu, T., Ha, T., Lokoc, J., Tim, Y.H.W., Joneja, A., Yeung, S.: Marine video kit: A new marine video dataset for content-based analysis and retrieval. In: Dang-Nguyen, D., Gurrin, C., Larson, M.A., Smeaton, A.F., Rudinac, S., Dao, M., Trattner, C., Chen, P. (eds.) MultiMedia Modeling - 29th International Conference, MMM 2023, Bergen, Norway, January 9-12, 2023, Proceedings, Part I. Lecture Notes in Computer Science, vol. 13833, pp. 539–550. Springer (2023). https://doi.org/10.1007/978-3-031-27077-2_42